DOCKET NO.
NAI1P022/01.106.01

U.S. PATENT APPLICATION

FOR AN

INTELLIGENT SPAM DETECTION SYSTEM USING STATISTICAL ANALYSIS

ASSIGNEE: NETWORKS ASSOCIATES TECHNOLOGY, INC.

KEVIN J. ZILKA
PATENT AGENT
P.O. BOX 721120
SAN JOSE, CA 95172

INTELLIGENT SPAM DETECTION SYSTEM USING STATISTICAL ANALYSIS

RELATED APPLICATION(S)

The present application is related to a co-pending application entitled "INTELLIGENT SPAM DETECTION SYSTEM USING AN UPDATEABLE NEURAL ANALYSIS ENGINE" which was invented by Anton C. Rothwell, Luke D. Jagger, William R. Dennis, and David R. Clarke, filed concurrently herewith under attorney docket number NAI1P023/01.152.01.

FIELD OF THE INVENTION

The present invention relates to SPAM detection methods, and more particularly to intelligently detecting and removing SPAM.

BACKGROUND OF THE INVENTION

The rapid increase in the number of users of electronic mail and the low cost of distributing electronic messages, for example, via the Internet and other communications networks has made mass marketing via electronic mail ("e-mail") an attractive advertising medium. Consequently, e-mail is now frequently used as the medium for widespread marketing broadcasts of unsolicited messages to e-mail addresses, commonly known as "SPAM."

Electronic mass marketers (also called "spammers") use a variety of techniques for obtaining e-mail address lists. For example, marketers obtain e-mail addresses from postings on various Internet sites such as news group sites, chat room sites, or directory services sites, message board sites, mailing lists, and by identifying "mailto" address links provided on web pages. Using these and other similar methods, electronic mass marketers may effectively obtain large numbers of mailing addresses, which become targets for their advertisements and other unsolicited messages.

Users of Internet services and electronic mail, however, are not eager to have their e-mail boxes filled with unsolicited e-mails. This is an increasing problem for Internet service providers (ISPs) such as America Online (AOL®) or Microsoft Network (MSN®) and other entities with easily identifiable e-mail addresses such as large corporations (e.g., IBM®, Microsoft®, General Motors®, etc.). ISPs object to junk mail because it reduces their users' satisfaction of their services. Corporations want to eliminate junk mail because it reduces worker productivity.

To date, the prior art has been devoid of mechanisms that can block SPAM effectively. Traditionally, SPAM detection has been based around specific rules for detecting it. Such rules include searching for key phrases in the subject headers, determining whether the recipient is actually on the list of users to receive the e-mail, etc.

More particularly, text search mechanisms are often used which rely on a centralized list of particular known strings. The strings on such list are usually specific trade names, products, sender, etc. As such, any variation in future spamming content results in a miss. Thus, what is needed is a process for detecting unwanted SPAM electronic mail messages in a more intelligent manner.

DISCLOSURE OF THE INVENTION

A system, method and computer program product are provided for detecting an unwanted message. First, an electronic mail message is received. Text in the electronic mail message is decomposed. Statistics associated with the text are gathered using a statistical analyzer. The statistics are analyzed for determining whether the electronic mail message is an unwanted message.

In one aspect of the present embodiment, the statistics gathered using the statistical analyzer include one or more of the following: a ratio of words capitalized to total number of words, a punctuation to word ratio, a number of URLs in the text, a number of (toll free) telephone numbers in the text, results of an analysis of a URL in the text, results of an analysis of e-mail addresses in the text, results of an analysis of character type (i.e. Unicode), and results of a message header field analysis.

The statistics can be placed in a results table. Entries in the table are passed as inputs to a neural network engine. Preferably, the statistics are compared to predetermined weights in the neural network engine for determining whether the electronic mail message is an unwanted message.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure **1** illustrates a network architecture in accordance with the one embodiment.

Figure **2** shows a representative hardware environment associated with the computers of Figure **1**.

Figure **3** is a flow diagram of a process for detecting an unwanted message.

Figure **4** depicts an illustrative architecture according to an embodiment.

Figure **5** is a flowchart of a process for teaching a neural network engine to recognize an unwanted message.

Figure **6** is a flow diagram depicting processing performed by the neural network engine.

Figure **7** illustrates a system for allowing a user to teach the neural network engine to recognize unwanted messages.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

Figure **1** illustrates a network architecture **100**, in accordance with the one embodiment. As shown, computers **102** of remote users are connected to a network **104**. The remote users send electronic mail messages (e-mail) to local users, who receive them on computers **106**. In the context of the present network architecture, the network may take any form including, but not limited to a local area network (LAN), a wide area network (WAN) such as the Internet, etc. The computers can include a desktop computer, laptop computer, hand-held computer, etc.

The e-mail passes through a gateway **108** which analyzes the messages to determine whether they are SPAM prior to allowing the message to pass to the local users. In one embodiment, the gateway **108** may include an Internet gateway, intranet gateway, Internet proxy, intranet proxy, or any other type of intermediate device. The gateway includes a statistical analyzer **110** and a neural network engine **112**.

In use, the statistical analyzer decomposes the electronic messages to determine an amount of various SPAM indicators (i.e. capitalization, punctuation, URLs, phone numbers, etc.). Instead of using static rule-based logic to accept or reject the electronic messages based on the results of the statistical analyzer, the results of the parsing are passed to the neural network engine. The neural network engine can be used in combination with the statistical analyzer to accept or deny electronic messages. An administrator **114** in communication with the gateway can be allowed to manipulate operation of the gateway and its components.

Figure **2** shows a representative hardware environment that may be associated with the remote source **102** and/or target **106** of Figure 1, in accordance with one

embodiment. Such figure illustrates a typical hardware configuration of a workstation in accordance with a preferred embodiment having a central processing unit **210**, such as a microprocessor, and a number of other units interconnected via a system bus **212**.

The workstation shown in Figure **2** includes a Random Access Memory (RAM) **214**, Read Only Memory (ROM) **216**, an I/O adapter **218** for connecting peripheral devices such as disk storage units **220** to the bus **212**, a user interface adapter **222** for connecting a keyboard **224**, a mouse **226**, a speaker **228**, a microphone **232**, and/or other user interface devices such as a touch screen (not shown) to the bus **212**, communication adapter **234** for connecting the workstation to a communication network **235** (e.g., a data processing network) and a display adapter **236** for connecting the bus **212** to a display device **238**.

The workstation may have resident thereon an operating system such as the Microsoft Windows NT or Windows/95 Operating System (OS), the IBM OS/2 operating system, the MAC OS, Linux or other UNIX operating system. It will be appreciated that a preferred embodiment may also be implemented on platforms and operating systems other than those mentioned. A preferred embodiment may be written using JAVA, C, and/or C++ language, or other programming languages, along with an object oriented programming methodology. Object oriented programming (OOP) has become increasingly used to develop complex applications.

SPAM DETECTION

Figure **3** is a flow diagram of a process **300** for detecting an unwanted message. In operation **302**, an electronic mail message is received. Text in the electronic mail message is decomposed in operation **304**. Statistics associated with the text are gathered in operation **306** using a statistical analyzer. In operation **308**, a neural

network engine coupled to the statistical analyzer is taught to recognize unwanted messages based on statistical indicators. The statistics are analyzed in operation **310** utilizing the neural network engine for determining whether the electronic mail message is an unwanted message.

According to another embodiment, a method is provided by which it is possible to accurately detect SPAM by applying statistical calculations against the text content of the message and supply the results of the application of the calculations and the message text contents itself into a neural network engine. The neural network then attempts to determine whether the message is SPAM or not based on what the neural network has learned in the past and/or by comparison with an existing set of known SPAM. An additional mechanism can be provided so that a user can return a message to the SPAM engine and mark it as SPAM (or not SPAM) to provide the engine with an on-going learning capability.

ARCHITECTURE

Figure **4** depicts an illustrative gateway architecture **400** according to an embodiment. The e-mail **402** arrives from the Internet **404** at the word decomposer **406**, which breaks the text content into words and punctuation. The parsed text is then supplied to the statistical analyzer **408** which creates a table of variables relating to the message, e.g.: Total Number of words, Number of words capitalized, Punctuation to word ratio etc. See Table **1** and related discussion, below.

This table along with the decomposed word list is supplied to the neural network **410** which provides a weighting, or probability, that the message is SPAM, partially based on known patterns of SPAM messages stored in a pattern database **412**. If the message is determined to be SPAM, it is quarantined in a quarantine database **414**. If

the message is determined not to be SPAM, the message is sent to a mail forwarder **416**, which forwards the message to a user **418**.

STATISTICAL WORD ANALYZER

The Statistical Word Analyzer attempts to build some key facts about the text content. The facts are based on certain characteristics that users/administrators have determined could represent SPAM. Such characteristics are set forth in Table 1.

Table 1

- Excessive capitalization
- Excessive punctuation
- Use of URLs and/or toll free numbers within text
- Text analysis by Unicode character type categorization
- URL analysis - checking for use of IP address rather than fully qualified name, and excessive numeric or symbol characters instead of alphabetic.
- E-mail address analysis - Checking for valid Internet addresses and excessive numeric or symbol characters instead of alphabetic. Also the absence of a From: or To: field or the presence of a large number of listed recipients in the case of an SMTP message.
- SMTP message header field analysis (Routing analysis for example)

Such characteristics are set forth in Example 1 which shows exemplary SPAM.

Example 1

To:
From:
Subject: $$$

There are other ways to make money!

This product produces 50% of all the money made on the Internet!
Now for the first time, it is brought to you retail!
People like you are making
$600-$4,000
per week in CASH with this product!
No selling! Not MLM! All CASH!!

Call Toll Free 1-888-555-6837 to find out more!!!

Only a few people per area will be selected to provide this revolutionary
product!
So act fast and be the first one in your area, and make the most money!!! Isn't
it time you earn
what you are worth? Aren't you tired of making someone else rich? Well, here is
your chance to
make YOU RICH!!!
No selling! Not MLM! All CASH!!

Call now Toll Free 1-888-555-6837 24 HRS!!!

Fortunes have been made with this product, and fortunes will be made again with
this new retail
version! Remember, get in at the beginning, the first ones in get the best locations!

Call Toll Free 1-888-555-6837 if all reps are busy, leave your name and number

```
and your call will
be returned in a few minutes!!!

Visit our website at:
http://192.168.3.1/93156483333954/~homepage/~john
```

A results table from analysis of the message of Example 1 is presented in Table 2.

Table 2

Number of whole words capitalized to total words ratio	3.3%
Total punctuation to word ratio	23%
% of ! in punctuation	69%
% of $	4%
% of ?	4%
URL properties: ▪ Number of URL's present. ▪ IP address instead of fully qualified domain ▪ Total numeric/symbol to alphabetic characters ratio (not including the IP address)	 1 True 50%
E-mail address properties: ▪ Sender specified ▪ Number of recipients ▪ Number of recipients falls outside of accepted boundaries (e.g. > 1 and < 5).	 False 0 True

At this point, the program segment shown in Example 2 may be used.

Example 2

```
If words_capitalized >5% and total_punc>20%
   And perc_plink >50% And URL analysis fails
   And E-mail address analysis fails Then
      This is spam
Else
      This is not Spam
End If
```

However, questions may arise as to whether the above analysis is accurate for all SPAM, whether the percentages are right, or whether the test variables are the correct ones to use.

Because this task is difficult to do using a fixed algorithm, the statistical analysis ends at this point and this data is passed to the Neural Network engine to determine patterns in statistics and words, and use these to determine whether the message is SPAM based on comparing the patterns to patterns predetermined to be SPAM or non-SPAM. The greater the number of variables in the statistics table, the easier it is for the Artificial Intelligence engine (AI) to "learn" to differentiate between SPAM and genuine messages.

The AI solution provides two goals. In particular, the AI is used to produce a set of rules that can be used in an existing AI engine. Further, the AI engine is used as a standalone gateway for determining which messages are SPAM.

Table 3 illustrates various steps and/or functions to achieving these goals.

Table 3

Pre-train the system with known SPAM from an archive of known SPAM, such as http://www.annexia.org/spam/ "The Great Spam Archive".
User Interface (UI) modification of the AI. A point and click UI where an existing e-mail is selected and displayed to be made an example of. Rules are constructed from the areas of the example mail that classify it as SPAM.

Application of a tiered approach, in which the user sends the message to an administrative area. An administrator utilizes the UI to notify the engine.

Artificial Intelligence introduced to make decisions based on previous Administrative input. This may include statistical or pattern-matching intelligence and would automatically update the rule-base.

Artificial Intelligence taken a level further, where grammatical and language decisions are made based on previous human-input to provide automatic generation of a rule-base.

System opened up to internal users with encryption method for trusted SPAM identification by the users.

NEURAL NETWORK ENGINE

The statistics table is passed as inputs to the Artificial Neural Network (ANN). The preferred ANN is a Supervised Learning type, though other types can be used. In this type, a "teacher" (user, administrator, or computer application) shows examples of inputs that the engine will receive paired with desired outputs. An error value is produced between the desired and actual responses, which should reduce as the learning progresses.

Figure **5** is a flowchart of a process **500** for teaching a neural network engine to recognize an unwanted message. In operation **502**, examples are provided to a neural network engine. The examples are of wanted messages and unwanted messages. Each of the examples is associated with a desired output. In operation **504**, each of the examples is processed with statistics for generating weights for the statistics. Each of the weights is used to denote wanted and unwanted messages. Logic associated with the

neural network engine is updated in operation **506** based on the processing by the neural network engine.

In the SPAM context, there are two sets of inputs: First, an archive containing only SPAM is inputted, and secondly an archive containing only genuine (non-SPAM) messages is inputted. Known SPAM can be obtained from various online resources (http://www.annexia.org/spam/ "The Great Spam Archive"). The teacher may automatically and randomly pick messages from either archive and supply them (with the statistical table) to the ANN together with a value for SPAM or non-SPAM. This continues until the archives are exhausted. A very large data set possibly consisting of over 500,000 messages is preferred.

Each input message is expressed as a vector, each variable from the statistic table being a feature variable in the vector. Note Table 4.

Table 4

$$X = \begin{matrix} \text{Capitals} \\ \text{Punctuation} \\ \text{Dollars} \\ \dots \end{matrix}$$

Figure **6** is a flow diagram depicting processing performed by the ANN. The most appropriate form of ANN would be an Adaptive Linear Combiner (ALC) **600**, which allows the presentation of input vectors X **602** and desired responses d **604** to the ALC. This will adjust weights until outputs a **606** are close to the desired responses.

After the learning process has taken place, the Engine can be deployed into the gateway situation. All associated vectors, matrices and weights to be used with the ALC can be stored permanently on disk.

The gateway could also be used to monitor intercommunication between Internet servers for tighter company-wide security, such as for preventing outbreak, SPAM, hacking attempts, etc. Such functionality can extend to the whole Internet community.

In addition to the pre-learning, there can also be an interactive learning mechanism while the gateway is active. This is discussed in the next section.

USER TEACHING MECHANISM

Some users of the system will be given permissions to allow teaching the Spam Engine when new mail arrives. Figure 7 illustrates the system that allows such teaching.

A typical sequence of actions using the system would be as follows. E-mail **702** is received by a user teacher **704**, but the e-mail has been marked incorrectly as clean. The user returns the e-mail **706** back to the Gateway **708** but indicates the desired response as "Spam." A learner **710** in the Gateway receives the e-mail and desired response. The Gateway stores the e-mail creating a unique ID for it. The Gateway sends a new e-mail **712** to the user. The new e-mail contains a unique ID contained within a URL to the Feature Editor (Java Applet) **714**. The user clicks on URL and indicates which features of the text of the e-mail make it SPAM. Preferably, the user is allowed to select the parts of the mail that clearly define it to be SPAM/offensive (e.g. subject, body, sender, attachments etc) and then within each part refine it down to the data elements to search on for CF filtering (set of words, file name / type for attachment, sender domain etc). The results of the user input are passed into a feature queue **716**.

The new features are not applied directly to the Statistical Analyzer/AI Engine **718** in one embodiment because the ANN would have to be reset and re-learn its mail

archive **720**. Instead, a batch teacher **722** schedules an entire learning run at a scheduled interval with additional features from the feature queue. A secondary ANN **724** is used so that the primary ANN can stay active and continue to process e-mail while the secondary ANN is learning. When the learning process is finished, the primary ANN becomes the secondary ANN and vice versa. The AI Engine is then initialized with the new features.

The AI engine could optionally be configured to divide SPAM into different confidence levels, e.g. Definitely SPAM, definitely not SPAM and possibly SPAM. The possibly SPAM would still be forwarded to the recipient, but also copied to a user teacher for feeding manual categorization in future training.

More information regarding the neural network engine may be found with reference to a co-pending application entitled "INTELLIGENT SPAM DETECTION SYSTEM USING AN UPDATEABLE NEURAL ANALYSIS ENGINE" filed concurrently herewith under attorney docket number NAI1P023/01.152.01, and which is incorporated herein by reference in its entirety.

While various embodiments have been described above, it should be understood that they have been presented by way of example only, and not limitation. Thus, the breadth and scope of a preferred embodiment should not be limited by any of the above-described exemplary embodiments, but should be defined only in accordance with the following claims and their equivalents.